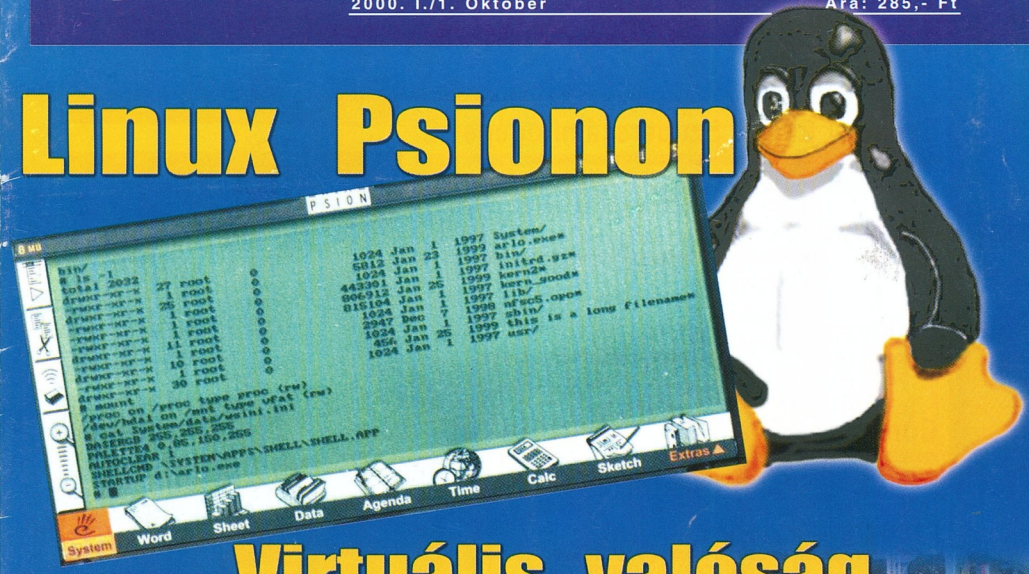
Számítógép felhasználók lapja

e-VILÁG

2000. I./1. Október

Ára: 285,- Ft



Linux Psionon

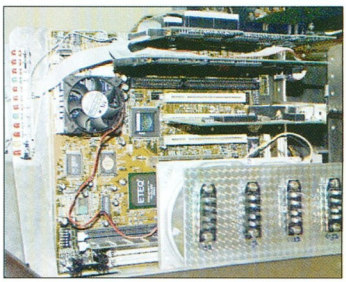
Virtuális valóság
nem csak játék

Beágyazva
QNX, RTLinux

Palmok kavalkádja

SZESZ:
kis megoldás nagy problémákra

Tartalom



Szegény Ember Szuperszámítógépe 4.

Álomgyár 8.

Beágyazva 10.

Corel Photo Paint 9 for Linux 13.

Linuxot a Psionra 15.

Revolúció 16.

Palm kavalkád 19.

Becsapható protokolok 20.

Linux PalmTop

Beköszöntő

Kedves olvasó!

Örömmel jelenthetem, hogy most már a magyar érdeklődők számára is rendelkezésre áll egy Linux témájú havilap. Ön most az első magyar Linux-al foglalkozó havilap első számát tartja kezében. A véletlen úgy hozta, hogy mindjárt két lappal is gazdagabbak lettek a Linux tábor képviselői. Talán velünk éppen egy napon (2000. október 7-ei Linux konferencián) mutatkozik be egy másik magazin is, a Linux Journal magyar kiadása. Mint Linux felhasználók örülünk, hogy végre magyarul is olvashatjuk ezt a színvonalas kiadványt. Reméljük, hogy az olvasók megelégedésére megfelelően sikerül kiegészítenünk egymást, és megszüntetni azt a hiányt ami ezen a területen eddig létezett.

Célunk bemutatni a magyar Linux világ szereplőit, továbbá ötleteket adni a felhasználóknak a különböző alkalmazási területek bemutatásával. Gyakorlati tanácsokat, tippeket is fogunk adni, igaz ez az első szám meglehetősen elméletire sikerült.
A Linuxon kívül nagy teret kívánunk biztosítani a PalmTop gépeknek is. Ezek a gépek a maguk területén legalább olyan hatékonyak mint a Linux és eddig ugyanolyan kevés figyelmet is kaptak.

Induló lap lévén szeretnénk bátorítani minden kedves olvasót, hogy írja meg véleményét, észrevételeit, kívánságait a lappal kapcsolatban. Igyekszünk a kívánalmaknak megfelelni.

Kívánom minden olvasónak, hogy teljen öröme a Linux használatában, és remélem, hogy egyre többen belátják, hogy pl. a szövegszerkesztés nem egyenlő fogalom a MS Word-el.

• *Farkas László*

Kiadja: Dikaro Bt. - 5435 Martfű Szolnoki út 91.
e-mail: diakro@westel900.net
Készült: Karcagi Nyomda Rt. Felelős vezető: Tankó Csaba
ISSN: 1586-5919

Előfizethető a kiadónál.
Fél éves előfizetés: 1.710,- Ft • Egy éves előfizetés: 3.135,- Ft
Bankszámla szám: CIB 10700309-25171008-51100005

Szegény Ember Szuperszámítógépe

Régóta ismert a kelet-európai programozók jó problémamegoldó képessége. Ez jórészt annak köszönhető, hogy a felmerült feladatokat jóval szerényebb eszközökkel kellett megoldaniuk, mint nyugati kollégáiknak. Erre jó példa lehet, amikor egy alkalommal egy hazai repülésirányító helyiségbe lépve, ott C64-esek sorakoztak. Ennek a múltban inkább a COCOM rendelkezések voltak az okai, manapság inkább az anyagiak.

Poor Man's Supercomputer (PMS)

A PMS projekt ötlete a Múzeum kert mögött lévő Favorit nevezetű pincehelységben született meg egy kis sörözés közben 1998 egyik márciusi délutánján. (Az volt néhányunk kis "irodalmi kávéháza" amig az ELTE nem költözött ki Lágymányosra.)

No nem ezért hívjuk néha SZESZ-nek az eredményül előállt számítógépet, hanem azért mert egyszerűen a magyar forditás (Szegény Ember Szuperszámítógépe) kezdőbetűi ezt adják ki.

Persze az ötletet nem a friss márciusi szellő hozta, hanem egy közvetlen fizikai problema: Einstein híres egyenlete szerint az energia átalakulhat tömeggé.Ennek fordítottja játszódik le akkor, amikor egy atom vagy anyagi részecske találkozik az "anti" párjával. Ekkor mindkettő megsemmisül, és az einsteini képletből kiszámolható energia jön létre. Ez eddig az iskolákban tanulható általános, szép, szimmetrikus fizikai világkép része. Az azonban mér sokkal kevésbbé ismert, hogy a fizikusok ezt a -számtalan kisérlet által bizonyított- tényt még nem tudták összhangba hozni a létezesünk (úgyszintén jól bizonyított) tényével. A probléma az, hogy az ismert fizikai egyenletekben az anyagot semmi sem tünteti ki az antianyaggal szemben. Ugyanakkor saját létezésünk bizonyítja, hogy valami fontos folyamat előnyben részesítette az anyagot az antianyaggal szemben, különben ugyanannyi antianyag létezne mint anyag. Ezek viszont előbb-utóbb találkoznának, és akkor egyből meg is semmisülnének. Így csak energia maradna az Univerzumban, és magunk sem létezhetnénk. Ha viszont az Ősrobbanáskor ("Big Bang") több anyag keletkezett mint antianyag, akkor a többlet "rendes" anyag megmaradhatott, és létrehozhatta az Univerzumot.

Ezen probléma tudományos vizsgálata az egyik tipikus felhasználása a világ leggyorsabb számítógépeinek. Ilyen gép viszont nincs Magyarorszagon, ezért az ELTE néhány elméleti fizikusa megpályázott egy u.n. APE szuperszámítógépet hazai kutatásaikhoz. Bár kiemelt támogatásban részesültek, a megitélt pénz csupán 32 PC-re volt elegendő. Amikor arról beszélgettünk a fent említett irodalmi kávéházban, hogy mit lehet ilyen helyzetben csinálni, akkor ötlött fel bennem a PMS gondolata. Később persze ezen még sokat kellett dolgozni. A terveket egyedül készítettem el saját szabadidőmben, de a furásban-faragásban az említett kollégákon kivül számos diák is segítkezett. Íly módon az Elméleti és Biológiai Fizikai Tanszék összefogásával elkeszült az első magyar tervezésű, PC alapú szuperszámítógép, a PMS1-et (lásd a címlap illusztrációt).

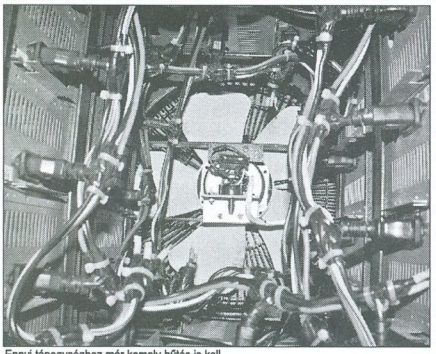
Itt hagy vessem mindjárt közbe, hogy a PMS az egy koncepció (mint pl. turbófeltöltés egy kocsiban.) a PMS1 pedig az első prototipus (az első autó ami ezen az elven működik). Igy tehát a PMS az fogalom, a PMS1 egy tárgy. Persze én azért is részesítem előnyben ezt a megkülönböztetést, mert a PMS1 sok olyan részlettel rendelkezik, ami nem feltétlen követendő minden PMS tipusú számítógépnél. Tehát valójában sokféle "PMS tipusú" számítógépet lehet csinálni. Pl. ha nem 32, hanem 128 (vagy N) PC-t teszünk bele. Ilyen apróságokon sokszor szabadalmak múlnak...

Természetesen felmerül a kérdés, hogy akkor mitől lesz egy számítógép PMS tipusú? Itt a legfontosabb annak megértése, hogy miben különbözik egy több processzorral rendelkező párhuzamos számítógép egy alaposan felturbózott, de egyetlen processzorral rendelkező számítógéptől. Az egyetlen processzort (és így az időmegosztáson alapuló irányelvet) tekintve szinte mindenki szakembernek tekinthető aki szereti a PC-s játékokat. A párhuzamos számítási módszerek (többprocesszoros hardware környezet) viszont még mindíg csak a "vájtfülűek" birodalmához tartozik.

Természetesen a világ minden (szuper- és nem szuper) számítógépe leírható "felturbózott, processzorhegyekkel és "terrabájtokkal" átütővé tett gép"-ként, mint ahogy egy modern Foma 1-es versenyautó is leírható a gőzerővel meghajtott ősautók "turbósított" változataként. Ebben az értelemben a turbósítás sok ötlet és szabadalom halmazának is tekinthető, hiszen az "őskomputert" Neumann János már megtervezte az 1940-es években. (Persze mondhatjuk, hogy végeredményben az is `csak' egy turbosított abakusz.)

Viszont a PMS teljesítményét például egyetlen -bármilyen módon felturbósított- PC sem tudná elérni (bár egy nagyon felturbózott PC-ért elkérhetnék akár a PMS árának felét ($10,000-t) is). Az ok nagyon egyszerű: a PC világban az "időmegosztás" a kulcsszó. Ez az ötlet Kemény Jánosnak (a BASIC nyelv megalkotójának) jutott eszébe, és nagymértékben "felturbósította" Neumannak az Eniac-nak elnevezett gépnél alkalmazott elképzelését.

Az időmegosztású gépeknél (pl. a PC-k) egyetlen processzor hajt min-



Ennyi tápegységhez már komoly hűtés is kell

den feladatot végre oly módon, hogy egymás után minden feladattal foglalkozik egy kicsit, és így a PC-khez képes hihetetlenül lassú emberek számára úgy tűnik, mintha "egyszerre" végezne el a PC minden feladatot. Azonban sok olyan feladat van, amit íly módon nem lehet elvégezni. Gondoljunk például el, hogy tíz embernek kell elvinni egy 10 mázsás szekrényt 10 méterre. A feladatot egy időmegosztásban végző csapat úgy próbálná megoldani, hogy minden embernek a 10 mázsás szekrényt el kellene vinnie 1 méterre, ahol a következő csapattag átvenné és tovább vinné 1 méterrel. Így persze egyetlen centit sem fog a szekrény előrébb mozdulni.

A PMS-ben a node-k párhuzamosan dolgoznak: ennek szellemében a fenti csapat mind a tíz embere egyszerre megfogná a szekrényt, és a ráeső egytizednyi súllyal a vállán elmennének egyszerre 10 métert. Ezt egy erős tagokkal rendelkező csapat minden további nélkül végre tudná hajtani. Ebben az értelemben, a PMS végtelenszer gyorsabb mint az ő nodjainak számával megegyező PC-kből álló független PC farm. Persze még az utóbbi -párhuzamos elven működő- "csapatnál" sem mindegy, hogy miként választjuk meg a csapattagokat. Akkor amikor manapság már 800MHzes processzor is kapható a piacon (Az 1Gigahertzes lényegében egy túlhajtott 800Mhz-es, és ezért kell egy fridzsidert is betenni ezek mellé), célszerű egy pillanatra megállni, és elgondolkozni azon, hogy érdemes-e az egyes nodokba a jelenleg kapható leggyorsabb processzorokat betenni. A párhuzamosság elve azt diktálja, hogy nem!

Tekintsünk egy egyszerű példát:

Egy 400 Mhz-es AMD processzor ára kb 42 USD, míg a 800Mhz-es 759 USD-ba kerül (www.pricewatch.com). Tehát egyetlen 800Mhz-es árából 18 olyat lehet venni, amelyek csupán fele olyan lassúak, thehát a 18 darab 400Mhz-es géppel 9-szer nagyobb számolási teljesítményt lehet elérni ugyanazért az árért (vagy ugyanazt a teljesítményt elérni 9-ed összegért). Ehhez persze a 18 processzort össze kell kötni párhuzamosan, és ez a kapcsolat időveszteséget jelent, de a számolás nagyságrendileg hasonló marad.

Természetesen a versenyfutás a szuperszámítógépek és a személyi számítógépek között örök: ha feltesszük, hogy az előző példában az egyes csapattagok tízszer erősebbé válnak a jövőben, akkor már akár egyetlen csapattag is meg tudja oldani a korábban -a rossz stratégia szerint- megoldhatatlannak tűnő feladatot; a holnap PC-i a mai szuperszámítógépek teljesítményét fogják tudni. A PMS-ben alkalmazott megoldás tehát arra szolgál, hogy már a jelenben is válaszokat kaphassunk a jövőben talán már egyetlen személyi számítógéppel is megoldható problémákra. És egyre több olyan probléma van, ahol a válaszadással nem várhatunk holnapig.

A PMS persze nem az egyetlen különleges számítógép.A múltban csináltak már a kutatók a legelképesztőbb elkepzeléseken alapuló u.n. speciális célú szuperszámítógépeket is. Viszont ezek mára szinte már kihaltak, mivel nagyon nehézkesen voltak programozhatóak, fejleszthetők. Ha az a diák aki az adott gép programozását diákéveisorán megtanulta elment máshova, a gép akár évekre is leállt. Közben persze a PC-k meg rohamosan fejlődtek, tehát egyre inkább szűkült az egyetlen PC és egy speciális célú szuperszámítógép teljesítménye közti rés.

A PMS nagy előnye, hogy a "csapattagok", azaz a számításokat végző egységek PC alkatrészekből állnak. Tehát, ha a PC ipar újabb és nagyobb teljesítményű alaplapokkal, processzorokkal áll elő, akkor azzal a PMS is egyre nagyobb teljesítményűvé válik. A node-k közti kommunikációt -azaz erőmegosztást- biztosító speciális hardware pedig mindíg alkalmazható az újabb modellekben is.

Térjünk vissza a PMS-ben alkalmazott konkrét egyedi megoldásra. A szuperszámítógépekben általában minden egyes processzorral közvetlenül el lehet érni egy tetszőleges másikat. A PMS azon tény felismerésének köszönheti létezését, hogy a legtöbb fizikai probléma megfogalmazható úgy, hogy csak az egymás mellett levő egységeknek kelljen egymással kommunikálni.

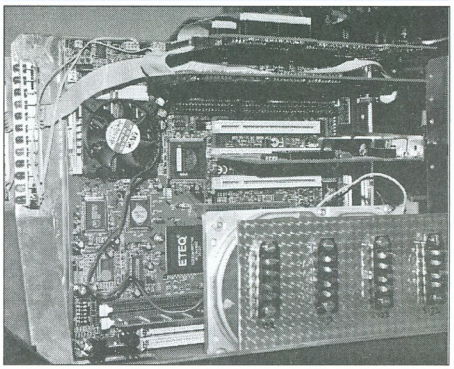
Hagy magyarázzam meg egy analógiával: A TV nézésekor megfelelő távolságból az ember egy összefüggő képet lát. Közelebb menve észreveheti, hogy a kép valójában egy rácsból áll. Ehhez hasonlóan a matematikai egyenleteket is rá lehet ültetni egy rácsra, és ha a rácspontok eléggé közel vannak egymáshoz, akkor a rácsra ültetett egyenlet pontosan úgy néz ki, mint az eredeti. Ha minden rácspontban ki tudjuk számolni az egyenletet, akkor végül is

az egész egyenlet megoldását kapjuk.

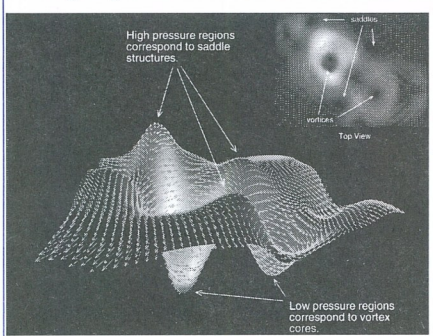
A TV képnél az egyes rácspontokban lévő pontok színe nagyon szoros összefüggésben van a szomszédainak a színével, de a TV két átellenes pontján levő pont színe majdnem független. A fizikai egyenletek rácson való megoldása is hasonló: ahhoz, hogy egy adott rácspontban kiszámoljuk az adott egyenlet értékét, ahhoz a legtöbb esetben elég ismerni a közvetlenül szomszédos rácspontokban az egyenlet értékét. Az ilyen feladatoknál teljesen felesleges az a hardver megoldás, amellyel egy adott processzor a rácsban elhelyezkedő többi processzor bármelyikét el tudja érni. Ehelyett öszpontosítani lehet a sebességet a legközelebbi rácspontokban levő processzorok között.

Ennek megvalósítására szolgálhatna bármilyen standard hálózat. Például az Ethernet (Token Ring) hálózatban egyetlen megszakítatlan kábel fut végig a PC-ken. Ennek eredményeként, ha két gép információt cserél egymással, akkor azok a teljes hálózatot lefoglalják. Egyszerre tehát csak egy pár kommunikálhat akárhány PC van a rendszerben. Ellenben ha pl. egy 32 PC-ből álló rácsnál minden modul csak a szomszédaival van közvetlen összeköttetésben, akkor itt egyszerre akár 16 pár is adatot cserélhet - a sebesség megsokszorozódik. Ez az egyik alapelve a PMS-nek.

Erre a célra létre kellett hozni, az állomásokat páronként dedikáltan összekötő adatátviteli csatornákat, melyet a teljesen új elven működő PMS kártyák valósítanak meg. Ezek nem kaphatók (egyelőre) kereskedelemben, és teljesen saját fejlesztés. Egy PMS kártya két részből áll; a processzor és az adatelosztó részből. A processzor rész tartja a kapcsolatot az alaplappal az adatbuszon keresztül, és továbbítja az adatot az elosztó résznek (vagy fordítva). Az elosztó rész feladata, hogy továbbítsa(vagy fogadja) a megkapott adotokat a megfelelő irányba (vagy irányból). Ha a PC-k egy háromdimenziós rácsban helyezkednek el, akkor minden állomásnak a hat szomszédjával kell a számítások közben adatot cserélnie, tehát az elosztó résznek hat független irányban kell tudnia az adatot küldenie vagy fogadnia. Természetesen lehet konfigurálni a rendszert úgy is, hogy csak két- vagy egydimenziós legyen. Ebben az esetben az adatelosztó résznek természetesen csak 4 illetve 2 csatornája kerül kihasználásra.



Az első PMS prototipus, az ELTE-n kifejlesztett PMS1 32 nodebol áll. A 32 modul mindegyike ugyanazon fő részekből áll: 450 MHz-es AMD K6-2-es processzorból, 4.1 Gbyte memóriából (128 Mbyte RAM egységenként), és 67 Gbyte háttér tárolóból(egységenként 2.1 GB HDD) áll. Minden modul tartalmaz persze PMS kommunikációs kártyát is. A processzoroknak külön tápegysége és hűtőventillátora van. Minden PC-re Linux operációs rendszer lett telepítve. Egy további számítógép vezérli az egész klasztert Ethernet hálózaton keresztül. A programok felmásolása és az eredmények lementése is ezen az Ethernet hálózaton keresztül történik. Az Ethernet tehát csak postásként van használva, a futtatás közben nem játszik szerepet.



Nagy előny, hogy nem kellett saját programozási nyelvet írni a géphez, mint a legtöbb szuperkomputer esetén. használható a hagyományos C nyelv. Mindössze két új utasítást kellett bevezetni, amit a C nem tartalmaz: a pms_küldés és a pms_fogadás utasításokat, amelyek az adatok küldését és fogadását végzik egy adott irányba.

Közvetlen kommunikáció csak a közvetlen szomszédok között lehetséges. (Ez azonban nem jelenti azt, hogy a távolabbi csomopontok teljesen el vannak vágva egymástól. Közvetve, két lépésben ezek is érintkezhetnek, de erre általában csak ritkán van szükség.)

Végezetül hagy említsem meg, hogy nemrég, amikor legutoljára Magyarországon jártam Chris Condon, a Business Week (http://www.businessweek.com/) európai újságírója megkérdezte,

hogy van-e stratégiám arra az esetre, ha olyan országokban is szeretnének PMS számítógépet telepíteni, amely országok nem nevezhetők igazán demokratikusnak.

A kérdés meglepett, mert addig fel sem merült bennem ez a probléma, de teljesen helyén való volt a kérdés: a teljesítmeny/ár tényező - amiben a PMS leginkább kimagasló - leginkább a feljődő országokban döntő tényező. (A legfejlettebb nyugati országok leginkább az abszolut teljesítmény kategóriában (a millió dolláros nagysagrendű szuperszámítógépekkel) versenyeznek egymással.) Ennek megfelel az a tény is, hogy Magyarországhoz keletről több érdeklődés érkezik mind emailben, mint pedig a PMS honnlapra mint a Dunához képest nyugatról. Ezért bár szívemhez közelebb van, hogy Romániától, Indián át Hon-Kongban is hozzákezdtek PMS tipusú gép építéséhez, igazán jó érzés, hogy már amerikai egyetemekről is komoly érdeklődés tapasztalható. A nem egészen demokratikus országok kérdésében pedig úgy döntöttem, hogy egyszerűen a Linux hagyományokhoz kötöttem a PMS terjesztését; ahol a Linux terjeszthető, ott szerintem a PMS is nyugodtan alkalmazható, és hasonlóan a Linuxhoz, a PMS architektúra és tervrajzok ingyenesen hozzáférhetőek nem profit orientált intézetek (egyetemek, kutatóintézetek, stb.) számára.

Egyébként ha már a PMS honnlapjáról volt szó, hagy említsem meg a címet:

`http://karman.phyast.pitt.edu/horvath/PMS/`

mely a híres magyar fizikusról, Karman Tódorról elnevezett szerveremen található. A magyar felhasználók részére hamarosan lesz egy könnyebben elérhető mirror honnlap is: `http://labor.elte.hu/horvath/PMS` címen.

■

Miért kell nekünk SZESZ?

Tekintsünk néhány érdekes példát:

• A Hegel-i gondolatat szerint nagyságrendi mennyiségi fejlődésen keresztülmenő rendszerekben új törvények jelennek meg. Ez így van az emberi civilizációval is. 60 évvel ezelőtt fele annyi ember élt a Földön mint manapság, és 20 év múlva kétszer annyi fog mint jelenleg. Ezt a mindennapok embere leginkább ott látja, hogy 60 évvel ezelőtt még az volt a gond, hogy miként szerezzen valaki kocsit, ma pedig, hogy hol parkoljon. Holnap pedig az lesz, hogy mivel tankoljon. Persze mondhatnám azt is, hogy 60 évvel ezelőtt a Dunából meritve még ihattak nagyapáink, ma már ez enyén szólva meggondolatlanság lenne.

Ezért a modern társadalmak törvényszerűségei mások mint korábban voltak és ezen új törvényszerűségek, vagy általánosan fogalmazva a "kollektív rendszerek viselkedésének" megértése egyike ezeknek a feladatoknak.

• Hasonlóképpen a környezeti áramlások is ilyen problémát jelentenek, mert amíg a Föld egy végtelen felületnek volt tekinthető, addig nem igazán számított, hogy az emberi civilizáció melléktermékei hova kerültek. Manapság a Föld egyetlen nap alatt bejárható veges és egyre szükösebb életteret jelent, és így nem mindegy, hogy a környezetkárosító termékek hogyan terjednek szét, merre vándorolnak a fizikai törvényszerűségeket követve. Ennek vizsgálatára nem elég egyetlen egyenletet betáplálni egy számítógépbe, mert a megoldáshoz azt is tudni kell, melyek a kezdeti- vagy peremfeltételek. Egy hasonlattal ahhoz, hogy megtudjuk mikor ér célba egy futó nem elég azt tudni, hogy milyen sebességgel halad, hanem azt is tudni kell, hogy mikor és honnét indúlt el, és pontosan milyen útvonalon halad.

Ezért az ilyen problémák számítógépes vizsgálata azt is megköveteli, hogy ideális esetben a Föld teljes felületének koordinátáit betápláljuk és egy adott pillanatban ismerjük a meterológiai viszonyokat ezekben a pontokban.

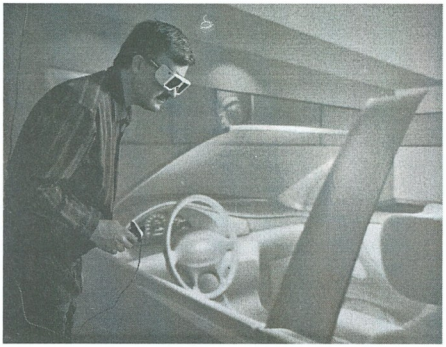
• A modern biológia is komoly számítógép teljesítmény igénnyel lép fel. Az emberi DNS teljes feltérképezése és visszakódolása olyan projekt, melynek megoldásától akár az átlagos emberi életkor megkétszerezéséét is várják a tudósok. Gondolom jónéhányunkat érdekel, hogy egy ilyen tudományos felfedezés nehány éven belül, vagy csak néhány száz év múlva történik meg (mert nagy valószínüséggel meg fog történni). Ez persze az előzőekben említett környezeti problémák megértését méginkább sürgetővé teszi.

• Az utolsó példa a szémítástechnika területéről van: A számítástechnikai kutatások egyik igen komoly területe az, hogy megállapítsák, hogy az USA-ban az Interneten használatos "128 bites kodoló rendszer", amely nélkül az Interneten veszélyes lenne vásárolni, mennyire biztonságos. Ennek kiderítése kulcskérdés; ha feltesszük, hogy ha akár 5 éven belül lesznek már olyan személyi számítógépek, melyek ezt a kódot meg tudják fejteni, akkor ma elég lenne csak háttértárolókra rögzíteni az amerikai kormány Interneten továbbított, mindenki által látható titkosított üzeneteit. Öt év múlva ezeket elővéve meg lehetne fejteni titkos 20-30 évre szóló startégiai információkat. Arról az apró körülményről nem is beszélve, hogy ennek a kódnak a közönséges számítógépekkel való visszafejthetősége azt eredményezné, hogy egyik napról a másikra tönkremennének a ma már teljesen általánosan működő, Interneten kereskedelmi forgalmat bonyolító cégek, bankok és egy általános amerikai tőzsdei katasztrófa magával ragadná az egész világgazdaságot. Ezért rengeteg pénzt fordít az amerikai kormány arra -és kifejezetten támogatja- hogy megállapítsák, hogy esetleg már a holnap személyi számítógépeinek feltételezett hihetetlen számolási teljesítményével ma rendelkező szuperszámítógépeken fel lehet-e törni ezt a kódrendszert.

• Én magam kétdimenziós hidrodinamikával foglalkozom. Az ilyen áramlások során a harmadik dimenzióban nem történik semmi érdemleges vagy azért mert a harmadik dimenzió sokkal kisebb mint a másik kettő, vagy azért mert egyéb fizikai hatások, pl. sűrűség különbség, ezt okozzák. Ilyen áramlásokra számos példa van. A Földet körülvevő légkör vastagságát és területet figyelembe véve az leginkább egy szappanbuborékhoz hasonlít! Ha ez a szappanhártya akadályokon (mint például magas hegyeken) bukdácsol keresztül, akkor örvények alakulnak ki. Az, hogy milyen sebességeknél alakulnak ki és tünnek el ezek az örvények, egyáltalán mi a kialakulás mechanizmusa, alapvető információ lenne a hurrikánok, tájfunok előrejelzésénél. Ilyen geográfiai kiterjedésű kisérleteket nyílván nem lehet elvégezni, ezért nekem abban fog segíteni a PMS, hogy jobban megértsem a kétdimenziós áramló közegekben kialakuló örvények dinamikáját.

Álomgyár

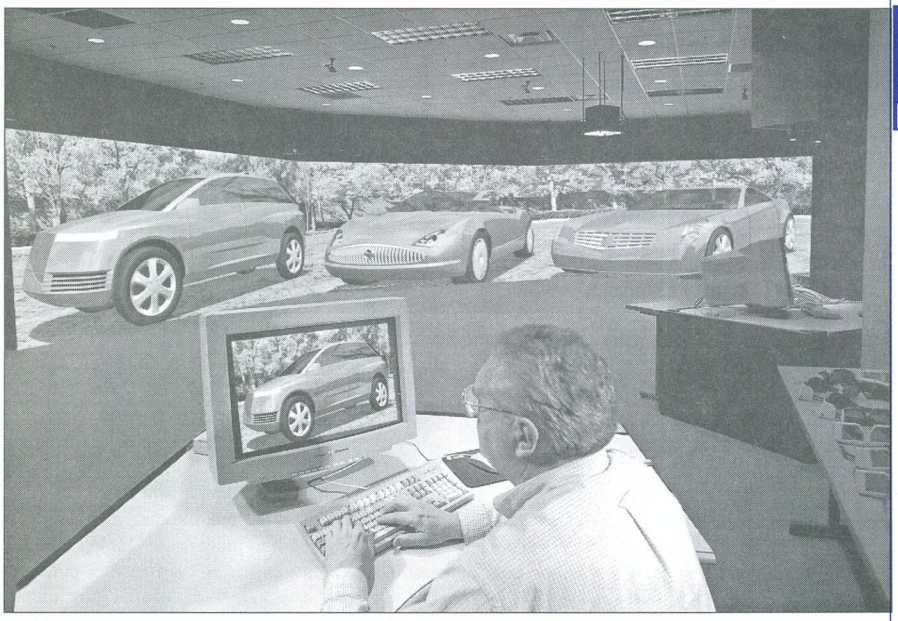
Az autótervezésben a számítástechnika nagy változásokat hozott az elmúlt években, ez jól látszik a drasztikusan lerövidült járműtervezési időkön. Manapság a kiállításokon bemutatott stílustanulmányból 2-3 éven beül sorozatgyártású autó válik.



A legfrissebb számítógépes technológiáról az autós szaksajtón keresztül is értesülhet a nagyérdemű. A történet szorosan kapcsolódik a minden év januárjában megrendezendő Detroit-i Motor Show-hoz. Ez a kiállítás a legnagyobb hagyományokkal rendelkezik a tengerentúlon, legfőbb jellemzője, hogy rengeteg tanulmányautót mutatnak be rajta. Célja e fantázia szüleményeknek kettős: a közönség elkápráztatása. A másik jóval gyakorlatiasabb, teszteltetni a nagyérdeművel a jövő formai, technikai irányvonalait. Tavaly novemberében már General Motors közszemlére bocsátotta azoknak a show autóinak a képét, amelyek a 2000.évi Detroit-i kiállításon megjelennek majd a színpadon. Az érdekessége ennek az volt, hogy ezek a járművek, (szám szerint 9) még a valóságban nem léteztek csak a számítógép képernyőjén fitogtatták erejüket. Igazából ez nem egy nagy hír, hiszen egy jobb grafikai programmal mindenki tud autót „rajzolni". A felhasznált program és gép tette igazán különlegessé az újdonságokat.

Az igen sok számítást igénylő autóiparban már a megjelenésük óta használják a számítógéppel segített tervezés és gyártási (CAD/CAM) programokat és az űrhajózásban először alkalmazott szuper teljesítményű Cray ipari masinákat (80-as évek közepe). A számítógépek úgy negyedszázada kezdtek elterjedni az autógyárakban. Először computerrel készített műszaki rajzok formájában jelentkeztek, a ma használatos digitális modellek a 90-es évek elején érkeztek. Jelenleg az autótervezéssel párhuzamosan folyó virtuális gyártástervezésről, azonos platformon működő programok hálózatáról beszélünk, amit e-engineeringnek (elektronikus tervezésnek) neveznek. A computerek megérkezésének az idején kb. 10-12 év telt el az első rajzok elkészülte és a sorozatgyártás megkezdése között. Jelenleg ez a szám a GM-nél lement két évre és tavaly kezdték el első olyan projektet, amiben ez az idő már 18 hónapra zsugorodott!! A gyorsulás oka a számítástechnikában rejlik.

A GM egész világot behálózó telephelyein, 9.200 munkaállomáson mérnökök ezrei dolgoznak a Unigraphics Solution programjaival. A szoftver cég az egyik piacvezető a fenti termékek területén, a vállalaton belüli kommunikációs rendszerekben és a gyártás szervezésben is. A nagy beruházást igénylő frissítést az amerikai mamutcégnél tavaly végezték el a fenti állomásokon mindössze 6 hét alatt. Így kerültek fel a gépekre a Version 15, Version 5 változatok, a belső kommunikációs, e-mail rendszert is tartalmazó IMAN is előnyére változott. A Lotus Notes lapokra épülő IMAN teremti meg azt a hálózatot, amely segítségével egy időben, egy terven végezhetnek különálló csoportok egymástól teljesen eltérő feladatokat. Van még a Unigraphics tarsolyában néhány termelékenység gyorsító program, közöttük a Járműfejlesztési eljárás névre hallgató, VDP-ként rövidített alkalmazás, amely harmadával csökkentette az autó utcára kerülésének idejét. Ebben a Hi-Tech birodalomban a Visual Eyes program a leglátványosabb elem a kívülálló számára, amit a fent említett show autók elkészítésénél is használtak. A szem előtt lévő virtuális szemüveg már nem jelent újdonságot a számítástechnikában, az ötoldalú, speciális anyaggal borított szoba viszont már igen. Ennek



megléte miatt tudják a mérnökök saját szemmel látni életnagyságban, tapasztalni a készülő jármű minden porcikáját, elemezni miként illik környezetébe. Ha a forma megfelel a kívánalmaknak, akkor a hálózaton el lehet küldeni szilárdsági vizsgálatra, virtuális szélcsatornába, gyártást megtervező helyekre, töréstesztekre. Ha ezek a dolgok elkészültek, akkor történik az első életképes prototípus elkészítése. Az egész eljárással sok időt takaríthatnak meg. A mai öldöklő harcban ez rendkívül fontos. A másik hasznos dolog egy vállalatnál, a jó hírnév. Ezt meglehet alapozni azzal, hogy a vásárló kiváló terméket kap és nagyon el lehet rontani azzal, hogy a vadonatúj autójával be kell fáradni tulajdonosnak a szervizbe mert kicserélnek rajta ezt azt, mert a kocsi nem úgy viselkedik kanyarban, nem annyira stabil és tsa. S ezek a visszahívások sajnos egyenes arányban szaporodn(t)ak a fejlesztési idők lerövidülésével és az első félév vásárlói elmondhatják, hogy saját maguk által finanszírozott teszten vesznek részt. Megnyugtatásul, ez nem mindenkinél van így és remélhetőleg csak az autóiparban létezik ilyen. Visszatérve a számítástechnikához, az iparban használatos gépek mellett a Unigraphics gondolt Windows alapú rendszerekkel és személyi számítógépekkel felvértezett kisebb vállalkozásokra is. Ők a piacvezető Solid Edge nevű programot tudják használni a gyártás és termék tervezéshez.

• Benedek Attila

Magyarországon a programokat (a Visual Eyes kivételével) az Intergraph Magyarország forgalmazza - http://www.intergraph.hu
A Unigraphics technológiájának alapja a háromdimenziós modell, melyet minden szakterület párhuzamosan használhat és módosíthat a termékfejlesztés során. Gyakorlatilag minden mérnöki szakterület moduljait tartalmazza. Multiplatformos, de a Linux verziót még nem vásárolhatjuk meg. A fejlesztés jelenleg teszt fázisban van.

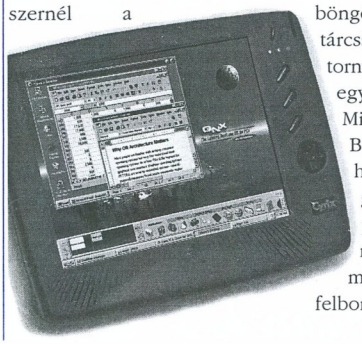
A Solid Edge Origin 2D CAD program CD-és változata most ingyen regisztrálható a http://www.intergraph.hu/ugsolutions/solidedge/origin web címen és még egy teljes kézikönyvet is letölthetünk mellé.

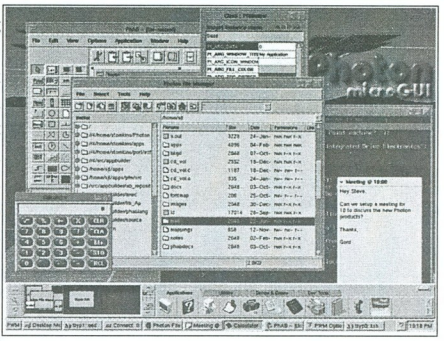
Beágyazva

Az operációs rendszerek piacán hosszú, csendes időszak után újra elindult valami pezsgés. Az asztali gépek területén a Windows továbbra is egyeduralkodó, és egyenlőre úgy tűnik, hogy a fölénye még egy jó ideig meg is marad. Ebben a csatában sokan szurkolnak a Linuxnak. Úgy tűnik, hogy a háborút a beágyazott rendszerek területén vívják meg a résztvevők. Nem olyan rég még leginkább csak az ipar területén találkozhattunk ilyen rendszerekkel. Manapság viszont már egy karórának is saját operációs rendszere lehet. A piac óriási és bár sok szereplős, még senki sem rendelkezik meghatározó fölénnyel.

A QNX Software Systems már közel 20 éve foglalkozik valósidejű operációs rendszer (RTOS) fejlesztésével. A valósidejűség beágyazott rendszereknél gyakran követelmény (lásd keretes írásunkat). A QNX Posix-kompatibilis, jól skálázható, moduláris felépítésű OS. Ez utóbbinak köszönhetően egyszerűen bővíthető az újabb igényeknek megfelelően. A modulok igen széles köre áll rendelkezésre. A modulok segítségével akár egy igen bonyolult, grafikus felületű rendszert is összeállíthatunk.

A modulok többsége újraindítás nélkül helyezhető üzembe. Egy beégyazott rendszernél a megbízhatóság követelmény. Képzeljük csak el, azt a PalmTopot amit lépten-nyomon hard-reszetelni kell, miközben adataink elvesznek. Ráadásul ezen rendszereknél sokszor nem is lehetséges az azonali beavatkozás, lévén nincs mellettük állandóan egy operátor (pl. egy műholdon lévő vezérlő egység). A QNX inteligens módon igyekszik helyreállítani a terminál előírt működési módját. Például ha valamelyik modul „lefagy" akkor azt automatikusan újra betölti és elindítja, miközben a rendszer többi része működőképes marad.





A méret a lényeg

Beágyazott rendszereknél a méret sem lényegtelen. A QNX-től igazán jó marketing megoldásnak bizonyult az a demo disk, ami weboldalukról (http://qnx.com) tölthető le. Segítségével egy bootolható flopy lemezt készíthetünk, mely egy QNX rendszert tartalmaz. Bootolás után a következőket kapjuk: teljes grafikus felületet, egy szövegszerkesztőt, egy grafikus file kezelőt, egy webböngészőt, természetesen egy tárcsázó programot, hanoi tornyai nevű logikai játékot, egy kis demo animációt. Mindez 1.44 MB-on!! Bootoláskor lehet kérni a hardwer detektálást. Minden gond nélül felismeri az alap perifériákat, megtalálja a modemet. Induláskor pedig már csak a kívánt képernyő felbontást kell kiválasztanunk. Ha beállítjuk internet szolgáltatónk adatait, (ami kb. 15 másodpercet vesz igénybe, semmi hokusz-pókusz), akkor a mellékelt böngészőt máris használatba vehetjük.

A mikrokernel alap kiépítésben mindössze 16 kernelhívást és három alap modult tartalmaz (ütemezés, üzenet továbbítás, interrupt átirányítás). A mikrokernel preemtive ütemezést használva több mint 2000 konkurens processzt képes kezelni, 32 prioritás szintre besorolva. Négy ütemező algoritmust használhatunk: érkezési, körbeforgó, alkalmazkodó és üzenet prioritáson alapuló (FIFO, round-robin, Adaptiv, message priority). Ez a kernel kb. 10 Kbyte méretű. Ehhez a kernelhez adhatjuk aztán a kívánalmainknak megfelelő modulokat.

A grafikus felület kialakításához a lehetőségekhez mérten három megoldás közül választhatunk. A Photon microGUI (ez található a demo disken is) a kis erőforrással rendelkező beágyazott rendszerekhez használható jól. Motifhoz és X-hez hasonló API felülettel rendelkezik. Teljes kiépítésben kevesebb mint 400 Kbyte, de már egy 256 Kbyte memóriát tartalmazó eszközt is képes ellátni egy asztali számítógéptől megszokott

grafikus külsővel. Másik megoldásként használhatjuk a legtöbb szolgáltatást nyújtó X Window rendszert. Ez igényli a legtöbb erőforrást. Harmadikként a kettő között helyezkedik el a QNX Windows, mely kevesebb szolgáltatást nyúj az X Windows-nál.

A QNX rendszer könyedén illeszthető különböző hálózatokba. A QNX saját hálózati protokollja, lehetővé teszi a hálózaton lévő gépek automatikus összekapcsolódását, a gépeken futó processzek egy közös ú.n. "message bus"-on keresztül kommunikálhatnak. Gépenként több hálózati kontroller támogatása, kontrollerek közötti hálózati terhelés megosztás a nagyobb átviteli sebesség és hibatűrés érdekében.

Windowsos környezetben a Phindows-t (Photon is Windows) használhatjuk. Segítségével a Windowsos desktopról indíthatunk QNX-es alkalmazásokat távoli rendszereken. Ezzel a megoldással elérhető, hogy a távoli rendszeren nincs is szükség fizikai képernyőre vagy billentyűzetre, mivel a megjelenítést vezérlő és az azt végrahajtható kód egymástól elválasztható. Ez a X-Window rendszerben dolgozó Linuxosoknak persze nem jelenthet újdonságot. Természetesen X Window alól is elérhetjük Photon-os alkalmazásainkat. Erre szolgál a Phinx (Photon in X Window). A fejlesztett beágyazott rendszer felhasználói felületét a Photon Relay segítségével jeleníthetjük meg távoli gépeken. Ez támogatja a legtöbb operációs rendszert és a legtöbb hálózati csatlakozást. Így a felhasználói felületet csak egyszer kell kifejleszteni, és használhatjuk különböző számítógépes környezetben.

Szeptember végétől magánfelhasználásra ingyen hozzáférhetővé tettek egy alapcsomagot (QNX Neutrino, Photon microGUI) néhány további kiegészítő csomaggal. Érdemes kipróbálni. http:// www. getqnx .com. A QNX Magyarországi disztribútora: ODD Kft. http:// www.odd.hu

*F.L.

Valósidejű operációs rendszerek

A valósidejű operációs rendszerek helye

A valósidejű operációs rendszerek a beágyazott rendszerek rohamos terjedésével egyre nagyobb teret nyernek. A hagyományos alkalmazási területeken, mint például ipari felügyelő rendszerek, repülés felügyelet, űrtechnika, intenzív őrző rendszerek, általában egy teljes képességekkel felruházott beágyazott PC kap helyet. Manapság a beágyazott berendezések egyre szélesebb körében váltják fel a valósidejű operációs rendszerek a korábbi egyszerű futtató rendszereket, ezáltal egyre gazdagabb alkalmazási lehetőségeket biztosítva ezen beágyazott berendezések számára. A legnagyobb alkalmazási területek az autó elektronika, mobil kommunikációs eszközök, szórakoztató elektronikai berendezések, háztartási elektronikai berendezések, stb. Ezen alkalmazási területeken az operációs rendszernek alkalmazkodnia kell a szűkös erőforrásokhoz. Ebből adódóan általában mikrokernel-architektúrát alkalmaznak, másrészt nagy hangsúlyt fektetnek a megfelelő erőforrás-gazdálkodásra.

Alapfogalmak

A valósidejű rendszerekkel és egyben valósidejű operációs rendszerekkel kapcsolatban érdemes tisztázni pár fogalmat. A valósidejű rendszereket nem azok nagy sebességű működése különbözteti meg az egyéb szoftver rendszerektől, hanem, hogy a valósidejű rendszerek egy (általában a külső környezetben) bekövetkezett eseményre egy előre magadott időkorláton belül garantáltan „válaszolnak". Ebből adódóan a rendszer helyes működése nemcsak a megvalósított funkciók helyességétől, hanem az előírt időzítések teljesítésétől is függ.

A valósidejű rendszerek két nagy csoportját különböztethetjük meg:

- kemény valósidejű (hard real-time) rendszerek
- lágy valósidejű (soft real-time) rendszerek

A kemény valósidejű rendszerek esetén az előírt időkorlát túllépésének katasztrofális következményei lehetnek. Ilyen rendszerek például az atomerőművek felügyelő rendszerei, a repülésirányítási rendszerek, a katonai védelmi rendszerek, az intenzív őrző rendszerek, stb.

A lágy valósidejű rendszerek esetén az előírt határidő túllépése esetén is értékes számunkra a rendszer válasza, azonban annak értéke az időkorlát megsértésének mértékével folyamatosan csökken, a rendszer degradált viselkedést mutat. Ilyen rendszereknek tekinthetők például a pénzkiadó automaták, a jegyfoglalási rendszerek, időjárás előrejelző rendszerek, stb.

Valósidejű operációs rendszerek

A valósidejű operációs rendszereknek az elsődleges feladata

- a hagyományos operációs rendszerekhez hasonlóan - az erőforrások, elsősorban a CPU, a memória és a kommunikációs hardver konzisztens kezelése. A hagyományos operációs rendszerekkel szemben viszont ez a funkció azzal a többlet teherrel bővül, hogy az operációs rendszernek figyelembe kell vennie a valósidejű feladatokat kiszolgáló folyamatok többletigényeit is, vagyis garantált válaszidőket kell biztosítania bizonyos események kezelésére. Ez jelentősen befolyásolja a kernel szerkezetet. A legjelentősebb többlet teher az ütemezésre hárul: a kernelnek valósidejű ütemezési algoritmusokat kell megvalósítania. A leggyakrabban használt valósidejű ütemezési algoritmusok: a prioritás vezérelt ütemezés, a határidő vezérelt ütemezés, az elegáns értékcsökkenéses ütemezés, az első határidő ütemezés és a minimális laxitás alapú ütemezés. Leggyakrabban egyszerű implementálhatósága miatt a prioritás vezérelt ütemezést alkalmazzák.

Itt érdemes megjegyezni, hogy az általános operációs rendszerekben előszeretettel alkalmazott FCFS vagy Round-Robin ütemezési algoritmusok alkalmatlanok valósidejű feladatok kiszolgálására.

A valósidejű feladatok kiszolgálása azonban speciális követelményeket támaszt az egyéb erőforrások kezelésével szemben is. Mivel garantált válaszidőket kell biztosítani, így az erőforrások kezelésénél olyan adatszerkezetek használata szükséges, amelyek karbantartása, a bennük való keresés ideje kézben tartható, korlátok közé szorítható. Tipikusan statikus erőforrás allokációt alkalmaznak, mert a dinamikusan allokált erőforrásokban a keresésre nem adható jól kezelhető időkorlát.

A valósidejű követelmények kezelésének megközelítései

Az operációs rendszer gyártók több megközelítés közül választhatnak, hogy az operációs rendszert alkalmassá tegyék valósidejű kritériumok kezelésére. Számos kereskedelmi valósidejű operációs rendszer azt a megközelítést követi, hogy az ipari igényeket kielégítő valósidejű kernelt bővítik általános, kényelmi szolgáltatásokkal. Ezt a megközelítést követi a beágyazott berendezések operációs rendszereinek piacán vezető szerepet betöltő VxWorks és bizonyos szempontból a beágyazott PC területen elterjedten használt QNX is.

Egy másik lehetséges megközelítés, hogy egy általános célú operációs rendszer kerneljét teljesen preemptívvé teszik, ezáltal eltüntetve a hosszú, meg nem szakítható kódutakat a kernelben, amik miatt ezen operációs rendszerek korábban nem feleltek meg a valósidejű feladatok kiszolgálására. Számos Unix rendszer követi ezt a megoldást: megvalósítják a POSIX 1003.1B szabvány szerinti, a valósidejű feladatok kezeléséhez definiált interfészt. A SOLARIS 2.x és az AIX is ezt az utat követi.

Számos operációs rendszer fejlesztő cég egy harmadik utat választ: egy teljesen a valósidejű követelmények teljesítésére optimalizált mikrokernelt használnak, amely alatt egy általános célú operációs rendszer fut, mint a legkisebb prioritású folyamat.

RTLinux

Egy általános célú OS. nem alkalmas valósidejű feladatok megoldására. Ennek oka egyszerűen az, hogy ezeket a rendszereket más feladatokhoz optimalizálták. Előfordulhat, hogy egy alacsonyabb prioritású processzre kell várakoznia egy magasabb prioritásúnak, hogy az „átadja" az éppen használt erőforrást. Ráadásul a válaszidők sem becsülhetőek meg nagy biztonsággal, ami szintén követelmény. Gondoljunk bele mi történne egy atomreaktorban, amikor a rendszer jelzést kap a teljesítmény csökkentésére, az éppen akkor kezdene heves „swapelésbe", és majd csak úgy 20mp múlva kezdene a feladat végrehajtásába (márha még létezik a reaktor).

Az RTLinux egy hard realtime operációs rendszer, ami a Linuxot, mint egy alacsony prioritású szálat futatja. A Linux szál végrehajtása teljesen preemtálható, így a realtime szálak és megszakítások sohasem késnek a nem realtime operációs rendszer (Linux) miatt. A megszakításokat először mindig az RT szál kapja meg és eldönti, hogy továbbadja-e azt a Linuxnak. A realtime szálak kommunikálhatnak a Linux processzekkel a megosztott memorián keresztül vagy egy file szerű interfészen, így a realtime alkalmazások képesek teljesen kihasználni a Linux nem-realtime szolgáltatásait is. (úgy mint hálózat kezelés, grafika, ablakozó rendszer, Linux készülékek és meghajtók, standard POSIX API) Például könnyű írni egy Perl scriptet ami Xwindows-on megjelenit adatokat, reagál a hálózaton keresztül érkező parancsokra, és összegyűjt adatokat egy real-time taskból.

RealTime kiegészítéseket írtak már a Windows NT-hez is, és ami elgondolkodtató, annak idején még az eleve RT-nek mondott Windows CE-hez is. Az NT mikrokernelje a fentiekhez mérten megmosolyogtató. Az RT Linux nem olyan rég leginkább még csak az egyetemek falain belüli próbálkozások eszköze volt. Jelenleg azonban már kinőtte ezeket a falakat. Bár hozzá kell tenni, hogy egy „valósidejűsített" rendszer sohasem tudja azokat a szolgáltatásokat ellátni, mint az eleve ilyen célra írt rendszerek.

RTLinux jelenleg 3 verzióban áll rendelkezésre:

RTLinux V1 egy kivételesen stabil rendszer. Egyaránt használható szállított termékeknél és laboratóriumi környezetben is. V1 RTLinux egy egyszerű API-t biztosít a realtime taszkok indításához, megállításához és ütemezéséhez.

RTLinux V2 a POSIX pthread API egyszerűsített változatát kínálja, ... A V2 kernel célja egy pontos, POSIX rendszert kínálni a sebesség és egyszerűség feladása nélkül, azzal teljesen kompatibilisen.

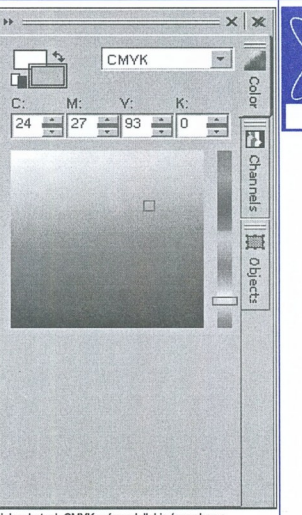
RTLinux V3 a rendszer PowerPC-re portolt változata. Béta tesztelése 2000 januárjában volt

Ezáltal biztosítható, hogy a valósidejű feladatok kiszolgálása mindig előnyt élvez az általános feladatokkal szemben. Az RTLinux is ezt a megközelítést alkalmazza. Ezen megközelítés hatékonyságát jól mutatja, hogy az RTLinuxot egyre elterjedtebben használják a valósidejű operációs rendszerek piacán. Érdekes megemlíteni, hogy egy RTLinuxot futtató PC irányította a Georges hurrikán „szemébe" adatgyűjtési céllal bevezérelt robotrepülőt is. Eredetileg az RTLinuxot is a beágyazott PC területén alkalmazták előszeretettel, azonban a beágyazott Linux megjelenésével a rendszer egyre jobban leskálázható, így alkalmassá válik a beágyazott berendezések korlátozott erőforrás viszonyai mellett is.

A valósidejű operációs rendszerek várható trendjei

A beágyazott berendezések piaca jelenleg az egyik legdinamikusabban fejlődő piac. Az előrejelzések ezen piac még dinamikusabb bővülését prognosztizálják. Másrészt a fejlesztési trendek azt mutatják, hogy a beágyazott berendezések egyre több erőforrással rendelkeznek, így egyre alkalmasabbá válnak mikrokerneles operációs rendszerek futtatására. Ezen berendezések funkciója és alkalmazási körének jelentős része megköveteli a valósidejű operációs rendszereket. A valósidejű operációs rendszer gyártók maximálisan alkalmazkodnak ezen piac igényeihez és valósidejű operációs rendszereiket jól skálázhatóra tervezik, hogy alkalmasak legyenek a beágyazott berendezések széles spektrumához. A jelenleg elérhető legismertebb valósidejű operációs rendszerek a teljesség igénye nélkül VxWorks, QNX, pSOS, pOSEK, RTLinux, iRMX, InTime, Windows CE, stb. A rendelkezésre álló valósidejű operációs rendszerek köre egyre bővül, egyre szélesebb körű szolgáltatások és magas szintű absztrakciókra épülő fejlesztőrendszerek várják a valósidejű rendszerek fejlesztőit.

• Dr. Román Gyula
BME, Méréstechnika és Információs Rendszerek Tanszék



dolgozhatunk CMYK színmodellel is (szemben a GIMP-el), így akár nyomdai előkészítésre is alkalmas a program.

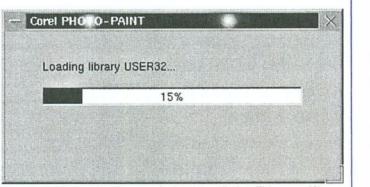
Tény, hogy gyorsabb, mint a Photoshop. A saját file formátumával. Viszont az egyéb gyakran használt grafikus formátumok mentése nem olyan olajozottan történik, mint a PhotoShopban. Gondolok itt pl. a jpg mentésre, melynél nagyon sok finomhangolási lehetőséget kínál fel, gyakorlattal nem rendelkező felhasználó ebben könnyen elveszik.

Nekem sohasem volt szimpatikus a PhotoPaint kijelölő eszköze. Könnyebben megy a PhotoShopban a kijelölés hozzáadása ill. levétele a Shift vagy a Control nyomva tartásával, mint a Photo Paintban állandóan nyomogatni az egérrel a pluszt vagy a minuszt. Végre a 9-es verzióban már ez a probléma

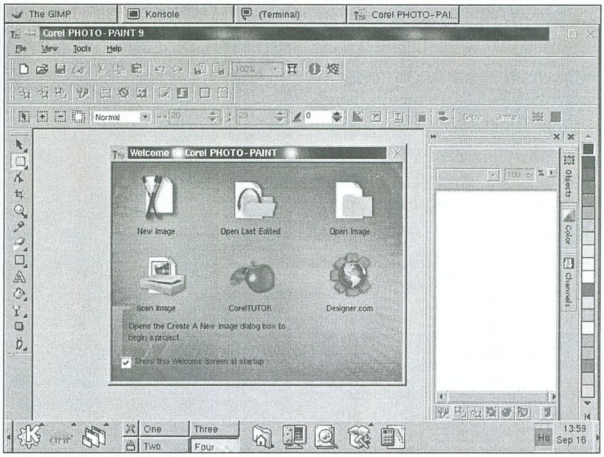
Corel Photo Paint 9 for Linux

A Linux használói igencsak bővelkedhetnek grafikai programokban. A Corel a közelmúltban örvendeztette meg a felhasználókat azzal, hogy ingyen hozzáférhetővé tette a Photo Paint9-et Linuxon. Elgondolkodtató, hogy egy kereskedelmi termékből, hogyan lesz szabadszoftver. A Corel talán így próbálja meg felmérni a piacot. Mindenesetre ez a termék, már alkalmas lehet a DTP-sek számára is.

A program sokat fejlődött az idők során. Régebben próbálkoztam használni, de akkor még annyira kezdetleges volt és kényelmetlen, hogy inkább maradtam a PhotoShop akkori verziójánál. Most, amikor feltelepítettem a gépemre, kellemesen csalódtam benne. Az eszközök kifinomultabbak lettek és talán több beállítási lehetőséget kínálnak, mint a Photoshopban. Némely pillanatban szinte már erőltetettnek érzem, látszik, hogy a nagy óriást utánozza, csábít, hogy térjek át erre a programra.

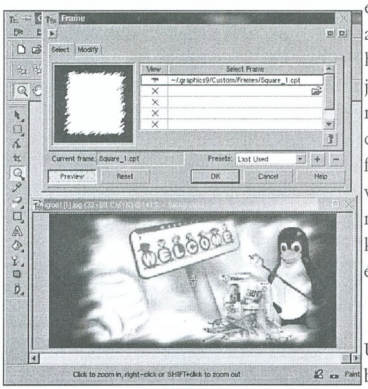


A program Linuxra ültetése nem forásszinten történt. Ehhez a saját célra kissé átírt Wine-t használták. A programindítás után ez indul el, és gondoskodik a megfelelő dll-ek betöltéséről. Emiatt az indítás több időt vesz igénybe mint Windows-on.



Indulás után a jól ismert képernyő fogad

nem áll fenn. Másrészt kedves a szivemnek, hogy a kijelölt rész tulajdonságai a megjelenő tulajdonság ablakban azonnal változtathatóak.

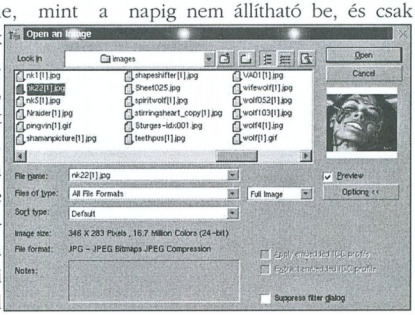


rengeteg effektből választhatunk

A rajzoló eszközök tárházát kínálja a PhotoPaint. Sokkal több eszköz van benne, mint a PhotoShopban. Sok fajta ecset, és ceruza. Szépen „fognak", könnyen lehet őket finomhangolni. Eszközökben a Photo Paint messze lekörözi a PhotoShopot, legalábbis a grafikus számára, aki sokféle eszközzel szeretne rajzolni, tervezni, új dolgokat létrehozni.

Igen-igen sok effektet pakoltak bele a Photo Paintba. Integráltak egy effekt csomagot is, ami a rossz minőségű képeket hivatott „feljavítani". Elég jó, de nem minden képnél működik korrektül, akárcsak a PhotoShop hasonló funkciója. A legtöbb képen valóban segít, de még mindig jobban szeretem kézzel beállítani a fényerőt és a kontrasztot.

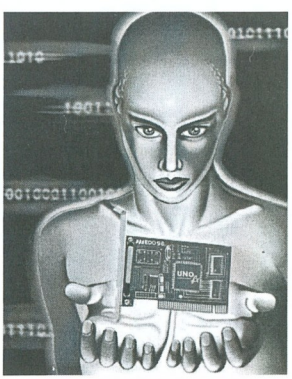
Elnyerte tetszésemet az Undo funkció, melynél beállítható, hogy ha hibát vét az ember akkor hányat lehessen visszalépni. A Photoshopban ez mind a mai napig nem állítható be, és csak egy hibát lehet visszavonni. Pedig néha jól jönne néhánnyal több Undo. Bár, a Photoshopban működik a History paletta, melyel visszavonhatók az egyes műveletek, de mégiscsak egyszerűbb lenne a Ctrl-Z-t nyomogatni, mint a PhotoPaint-ben..



Layereket a Photo Paint is kezel már. Úgylátszik a fejlesztők rájöttek, hogy ez igen hasznos dolog. Bár a rétegkezelés még egy kissé nehézkes a PhotoShop után.

Összefoglalva, a Photo Paint a Photo Shop babérjaira tör, és alaposan meg is szorongatja. Lényegesen több beállítási lehetőséget kínál, gyorsabb és majdnem olyan kényelmes, mint a nagy óriás. A gyorsaság itt meglehetősen objektiv dolog. Az effekteket gyorsan kiszámolja, de pl. egy file megnyitáskor, ha a preview funkció is be van kapcsolva akkor iszonyúan lasú. Számomra a széles eszköztár az, amit irigyelek tőle, és szeretném ha a PhotoShopban is benne lenne. Azt hiszem, ha a két program előnyös tulajdonságait összegyúrnánk, akkor kapnánk meg azt a képszerkesztő programot, ami minden grafius álma.

• *Körmendi Edit*
http://www.externet.hu/edo



Linuxot a PSION-ra

Linux7k

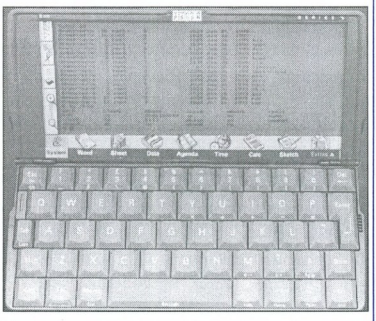
Ennek az érdekes projectnek a célja a Linux portolása ARM7110-es processzora illetve Psion 5mx architektúrára. Az ötlet első hallásra izgalmasnak tűnik, de utána rögtön felmerül a kérdés: mire jó ez?
http://www.calcaria.net

A kernel standard Linux kernelen alapszik. Támogatja az összes Psion specifikus készüléket, beleértve az érintésérzékeny képernyőt is. Sajnos ez a kernel még csak a Series 5-ös gépekre készült el. Ennek egyik oka lehet, hogy az 5mx architektúrájáról keveset lehet tudni, bár a fejlesztők szerint a Psion támogatja a projectet, és technikai segítséget is nyújt.

A Linux kernel betöltése nem is olyan egyszerű hiszen a Series 5 már rendelkezik egy operációs rendszerrel ami a ROM-ban van. Ez a kezdeti boot során azonnal el is indul, majd továbbra is fut, még akkor is amikor a gépet kikapcsoljuk. Valójában a gép nem is kapcsol ki teljesen, csupán egy alacsony energiaállapotú, készenléti módban működik.

A kernel betöltését az ArLo (ARM Loader) program valósítja meg, valahogy úgy mint a Loadlin DOS alatt.

Amikor az ArLo elindul, betölti a kernel image-t a memóriába és opcionálisan létrehoz egy RamDisk image-t is. Majd megszünteti a processzor védett módját és a kernel boot kódjára ugrik. Innentől már a Linuxé az irányítás. Felülírja önmagával a memóriát, melyet azelőtt az EPOC használt. Mivel ezt a szituációt nem láthatták előre a Psion tervezői, ez szintén okoz egy ki problémát. Ha újra indítjuk a gépet (hard-reset) miután már a Linux fut, automatikusan az EPOC fog elindulni (mivel a boot vektora firmware-be van ágyazva). A boot vektorba el van helyezve egy kód, mely azt figyeli, hogy történt-e már bootolás az utolsó hard-reset óta. Tehát elindul az EPOC, észleli, hogy már ezelőtt futott és teljesen összezavarodik, ahogy megpróbálja használni azt a memória tartalmat, amit még a Linux betöltése előtt használt, aminek jó részét a Linux már felülírta. Hogy ezt elkerüljük, a reset gomb lenyomása után, mialatt a gép újra indul, lenyomva kell tartani mind két shift billentyűt, ami arra kényszeríti az EPOC-ot, hogy újrainicializálja a memóriát.



Természetesen minden Linuxos alkalmazást újra kell fordítani ha használni akarjuk a 7K alatt is. Nagy valószínűséggel azonban ez még nem elegendő. Minden bizonnyal a forráskódot is át kell írni. Például a Psion kijelzője teljesen más formátumú mint egy asztali monitoré. A Linuxos alkalmazások elég nagy területet foglalnak el a képernyőből (pl. a KDE felcicomázott them-jei), míg az EPOC-on a jobb helykihasználás végett a menük is bezáródnak, ha éppen nincsenek használatban. Szerencsére az X Window grafikus motorja majdnem korlátlan konfigurálást tesz lehetővé. Így megvalósítható egy kimondottan Psionos ablakkezelő. Ráadásul az X Window rendelkezik virtuális desktop-pal, amik teret biztosíthatnak a Psion korlátozott megjelenítőjén.

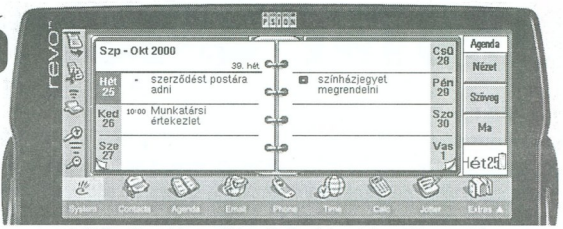
Mindez nagyon szép, de ki akarná lecserélni az EPOC-ot Linuxra? Az EPOC stabil, rendelkezik grafikus felülettel, és kimondottan plamgépekbe és más beépített rendszerekre tervezték. Jól el van látva felhasználói programokkal is. Akkor hát miért „üssük fejbe" Linuxal?

A K7 fejlesztői szerint sokkal több program érhető el Linuxon, és ezek fejlesztésére is több eszköz áll rendelkezésre. Ez kétségtelenül így van, de ezek az alkalmazások jóval több erőforrást is igényelnek mint amivel jelenleg egy plam gép rendelkezik.

Mindenesetre rendszeradminisztrátoroknak azért hasznos lehet. Segítségével akár a horgász-tó partjáról is bejelentkezhetnek a munkahelyi szerverre és elvégezhetik a szükséges beavatkozást, és ehhez nem kell táskányi berendezést magukkal cipelniük.

•

Revolúció



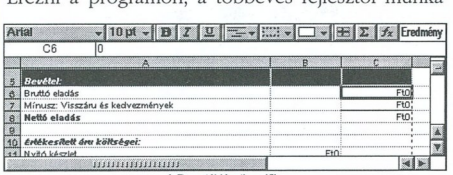
a határidőnapló (agenda) nagyon jól használható segítség a mindennapjaink szervezésében

Alig egy éve, hogy megjelent a piacon a PSION REVO készülék, máris itt (szept. 5.) az újabb változata a REVO PLUS. Elődjéhez képest nem tartalmaz sok újdonságot, inkább azok kedvében kíván járni akik utazásaik során sem akarnak (vagy nem tudnak) lemondani az interneten való szörfözésről. A memóriát megnövelték 8MB-ról 16MB-ra és mellékeltek hozzá egy WAP böngészőt valamint a népszerű Opera web böngészőjét.

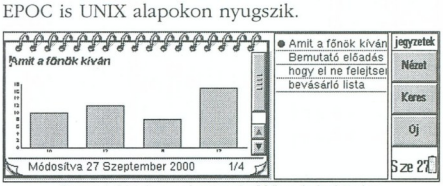
A REVO lelke egy ARM 710-es processzor, EPOC32-es operációs rendszerrel. Ez megegyezik a nagytestvérénél (Series 5mx) alkalmazottal. A beépített szolgáltatások és programok is jó részt megegyeznek. Amik kimaradtak: sketch rajzprogram, hangrögzítés, hangállományok szerkesztése, háttérvilágítás. A rajzprogram talán sokaknak nem jelent nagy veszteséget, akinek mégis hiányozna, annak sem kell nélkülöznie. Freeware programként beszerezhető. Sokkal fájdalmasabb a háttérvilágítás elvesztése. Igaz cserébe egy jobb minőségű, élesebb képet adó kijelzőt kapunk. A másik nagy különbség az 5mx-el szemben, hogy a REVO nem bővíthető. Azonban egy átlagos felhasználó számára már a 8MB is kinőhetetlennek látszik. Windowshoz szokott felhasználók ezt kétkedve fogadják. Egy 15.000 szavas szótárprogram pl. 85 Kbyte, egy sakkprogram help file-al együtt 140 Kbyte, egy 500 soros, 6 oszlopos táblázat 26 Kbyte, Verne Gyula, 80 nap alatt a Föld körül regénye 383 Kbyte.

Aki egyszer használatba veszi, nehezen lesz meg nélküle. Az elérhető alkalmazások nagymértékben képesek segíteni a napi munkát. Sok esetben egy laptop minden további nélkül kiváltható segítségével. A beépített alkalmazások közül (partner nyílvántartás, gyorsjegyzetelő, szövegszerkesztő, táblázatkezelő, határidő napló, egyszerű adatbázis kezelő, e-mail kliens, SIM kártya kezelő, világóra) a határidőnaplót (agenda) érdemes kiemelni. Érezni a programon, a többéves fejlesztői munka eredményét. Nem hagytak ki belőle olyan apróságokat sem, ami más hasonló termékből sokszor kimarad. Például az el nem végzett teendőink átcsúsznak a következő napra, így biztos, hogy nem marad el egy fontos teendő sem. Ezeket külön listákba szervezhetjük a könnyebb áttekinthetőség kedvéért. Választhatunk napi, heti, havi nézetek közül, és kérhetünk elfoglaltsági táblázatot is, könnyítve a másokkal való időpont egyeztetést.

A REVO-t természetesen csatlakoztathatjuk asztali gépünkhöz is. Ehhez telepítenünk kel a PsiWin kommunikációs programot (Macintoshon a PsiMac-et). Telepítés után úgy látjuk a csatlakoztatott Revonkat mint egy meghajtót. Böngészhetünk a file-ok között ill. oda-vissza mozgathatjuk azokat. A PsiWin megoldja a konvertálást is. Így ha a Revoról egy szöveges dokumentumot viszünk át az asztali gépre, akkor az automatikusan Word formátumban kerül át (ha a beállításoknál ezt kérjük), illetve ha egy word dokumentumot másolunk a Revora, akkor az szintén felveszi a megfelelő formátumot. Érthetetlen módon a kommunikációs programnak nics Linuxos változata, pedig maga az EPOC is UNIX alapokon nyugszik.



A Revo táblázatkezelője



Gyorsan és kényelmesen visszakereshető feljegyzések a jotter-ben.

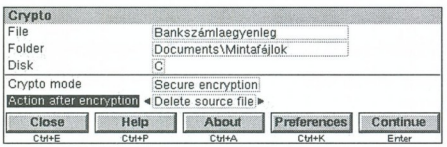
A billentyűzet nagyon jól sikerült. Egy ilyen kis készüléknél ez kritikus pont szokott lenni. A kis méret következtében a billentyűzet könnyen használhatatlanná válhat. A PALM-ba feltehetőleg éppen ezért nem is építettek. A Revonál viszont, ha nem is lehet tíz újjal gyorsgépelni, azért folyamatos gépelésre alkalmas maradt.

A Revon nem tudunk OPL programokat sem irni. Az ehhez szükséges alkalmazásokat szintén kihagyták (az 5mx-nél megtalálható). Szerencsére mégsem kell nélkülöznünk. A Symbian oldaláról a fejlesztő eszköz regisztrálás után letölthető. Az asz-

tali gépen tudunk programokat írni és a fejlesztő eszközhöz mellékelt emulátoron tesztelhetjük is az eredményt. Majd a végleges programot áttölthetjük a Revora. Általában is ez a fejlesztés menete, hiszen egy hosszabb programlistát könnyebb az asztali gépen szerkeszteni, mint a PalmTop kis képernyőjén. Ha már a letöltésnél tartunk, érdemes szétnézni a hálón. Rengeteg freeware vagy némi korlátozással, de használható shareware programot találhatunk.

Crypto • http://salvis.com/

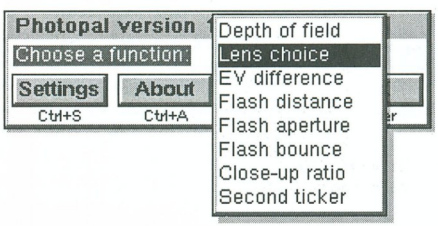
A Psion palmokon sajnos nincs file szintű védelem (kivéve a word és a táblázatkezelő alkalmazásokat, de ezek sem túl hatékonyak). A gép bekapcsolásához ugyan rendelhetünk jelszót, de ez egy idő után meglehetősen bosszantóvá tud válni. Ha pl. gyorsan szeretnénk megnézni egy ismerősünk telefonszámát, előbb a jelszóval kell bajlódnunk. Ezen a problémán segít a Crypto (freeware).

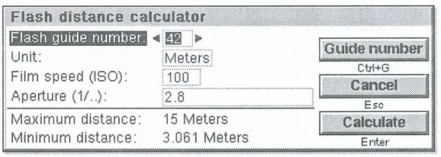


Segítségével bármely file-t megvédhetjük az illetéktelenek tekintetétől. Választhatunk, hogy csak jelszóval védjük a fileunkat, vagy titkosító algoritmust (256 bites RC4 algoritmust) is használjon a program. A csak jelszóval való védelem természetszerűleg a leggyorsabb. Egy titkosított file megnyitásakor elindul a crypto is. A jelszó bekérése után automatikusan dekódól (ha kértük az encryptálást is). A file-t ezekután módosíthatjuk. Bezárás után pedig a Crypto szintén automatikusan gondoskodik az újabb file védelméről. Egy már létező file titkosításakor egy új file jön létre. Ilyenkor ne felejtsük el törölni az eredetit!

Photopal • www.PocketIQ.com

Amatőr és profi fotósok vehetik hasznát ennek a kis freeware programnak. Segítségével a fotózáshoz szükséges paramétereket számíthatjuk ki. Úgymint például mélység élesség, vaku hatótávolság, szükséges rekesznyílás vakuzáskor, stb...

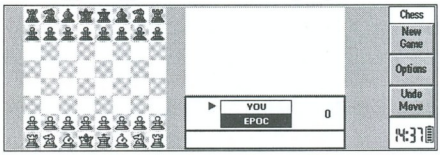




Ha ismerjük a megfelelő összefüggéseket, képleteket, akkor a táblázatkezelőben (jobban szeretem a számolótábla kifejezést) magunk is készíthetünk hasonlót. A program ad egy kis segítséget azon amatőr fotósoknak is, akiknek a sötétkamrájukban nem áll rendelkezésre exponáló óra. A másodperc számláló opció (second ticker) kiválasztásával a gép rövid hanggal jelez minden másodpercet. Így a sötétkamrában is számolható az idő múlása.

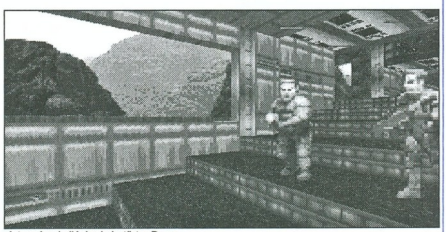
PocketChess for EPOC • http://www.eskimo.com/~scottlu

Az egyik legjobb freeware sakkprogram.



A nehézségi szint több fokozatban állítható. A nagyobb fokozatok csak a nagyon sok idővel rendelkező játékosoknak ajálható. De mivel az Epoc több feladatos OS. nem kell megvárnunk a válaszlépést. Dolgozhatunk tovább a többi alkalmazással, időnként pedig átválthatunk megnézni, hogy lépett-e már a gép.

•



A tenyérnyi világba is betört a Doom.
Igaz ez csak a Series 5mx-en fut. Mivel a Revo és az 5mx hardver szinten majdnem azonos érdemes kipróbálni az 5mx-re írt programokat is. Nagy részük elfut a Revon is. A kijelző mérete jelentősen eltér a két gépen, így az olyan alkalmazásokat, melyek az 5mx kijelzőjére építenek, mint a Doom is, sajnos Revon nem használhatóak (még ha el is indulna). Önmagában talán nem is túl élvezetes ez a játék palmgépen, de a beépített IrDa segítségével kábelezés nélkül, minden további nélkül játszhatunk mások ellen is. (még jól jöhet egy unalmas értekezleten)

Előfizetési nyeremény akció!

Egy és féléves előfizetőink között egy PSION REVO kéziszámítógépet sorsolunk ki.

Előfizetés

fél évre: 1.710,- Ft (6 szám)

egy évre: 3.135,- Ft (12 szám, 1 szám ingyen)

Növelje nyerési esélyeit!

Ajánlja másnak is a lapot! Ha az illető előfizet a lapra, ön küldjön nekünk erről egy feljegyzést (levélben vagy e-mail-en). Minden előfizető után, akinek Ön ajánlotta az újságot egyel növekszik az esélye arra, hogy megnyerje a Psion Revo kéziszámítógépet.
sorsolás: 2000. november 30.

(Az egymásra történő kereszthivatkozásokat kiszürjük és az illetőt kizárjuk a sorsolásból)

Saját előfizetésem

Név: .....

Számlázási cím: .....

Posta cím (ha eltér a számlázási címtől): .....

e-mail: .....

Barátom előfizetése

Név: .....

Számlázási cím: .....

Posta cím (ha eltér a számlázási címtől): .....

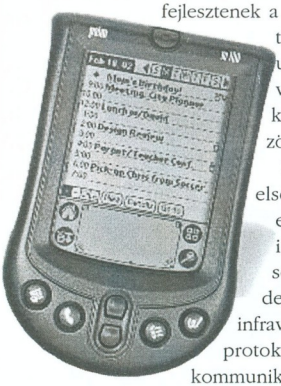
e-mail: .....

Beküldési cím:
e-VILÁG, 5435 Martfű, Szolnoki út 91.
e-mail: diakro@westel900.net

PALM kavalkád

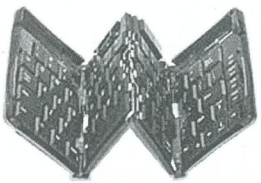
Míg Európában a Psion kézi-számítógépeké a vezető szerep addig az USA-ban a PALM sorozat tagjai a kedvencek. Magyarországon még keveset hallani felőlük, pedig már másfél éve hivatalosan is forgalmazzák itthon. Azt már megszokhattuk, hogy Magyarországot nem tekintik komoly piacnak, de a PALM-nál még Nyugateurópa is csak a harmadik helyen szerepel. Az USA-ban rengeteget adnak el belőlük. Sikerének egyik oka lehet a nagy tipusválaszték. Mindenki megtalálhatja az igényeinek (és persze pénztárcájának) legmegfelelőbb változatot. Létezik olyan tipus ami alig több egy manager calculátornál és van már színes kijelzővel ellátott tipus is.

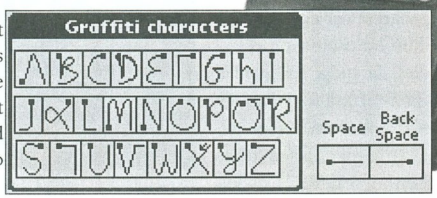
Az asztali géppel kapcsolatot tartó programnak (ez végzi a file konverziókat és gondoskodik az adatok szinkronizálásáról) a PALM gépeknél is csak Windowsos vagy Macintoshos változata létezik. Azonban a fejlesztőeszközök és az emulátor elérhető Linuxra is. A gépek nagy népszerűsége ösztönzi a fejlesztőket is. Rengeteg programot fejlesztenek a különböző PALM tipusokra és ugyanígy nagy a választék az egyéb kiegészítő eszközökben is.



Ezek a gépek elsődlegesen az elektronikus határidőnapló vezetésére alkalmasak, de támogatják az infravörös és a TCP/IP protokolon keresztüli kommunikációt is. Természetesen saját operációs rendszerrel rendelkeznek. A Palm OS sajnos nem multitasking rendszerű, így egyidejűleg csak egy alkalmazás futtatását teszi lehetővé.

Első kézbevételkor a billentyűzethez szokott felhasználónak furcsa a billenytűzet teljes hiánya. Az érintés érzékeny képernyőre közvetlenül írhatunk. Betaníthatjuk a Palmot írásunk felismerésére *(lásd az ábrát)*. Ez rövid szövegeknél jól működik, de hosszabb használatnál nehézkessé tud válni és persze a folyamatos kézírás nem felismerhető. Ezen a gondon segít át a hordozható billentyűzet. A billentyűzet összehajtható. Így viszonylag kis helyet foglal, de zsebbe sajnos még így sem tudjuk rakni. Kihajtva egy normál méretű eszközt kapunk. Így az a furcsa helyzet áll elő, hogy a csatlakoztatott készülék jóval nagyobb mint az eszköz, amihez csatlakoztattuk.





Becsapható Protokollok

A tisztán szövegalapú protokollokat mára már nem tanácsos használni olyan helyeken, ahol a biztonság kicsit is fontos. Ilyen protokollok a telnet és az FTP is. Ezen protokolloknak nem a tervezése rossz, hiszen az internet sokkal biztonságosabb volt, amikor kitalálták őket. Manapság viszont nagy könnyelműségnek számít ezen protokollok használata. Komoly támadásoknak tehetjük ki magunkat.

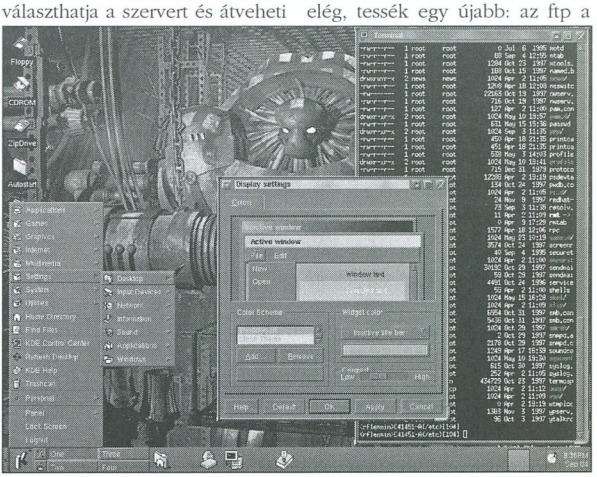
Először is nézzük a standard telnet session-t. Tudjuk, hogy az egész csomagokból áll, melyek az egyik gépről vannak átküldve a másikra az Interneten. Amikor pl. otthonról jelentkezünk be az irodába, akkor közbeiktatott host gépet is érintünk. Először az otthoni gépem csomagokat küld egy szomszédos router-nek, mely router továbbítja azokat egy másik router-nek, ami továbbküldi azokat az egyik ISP-nek. Az ISP-ről a munkáltatói ISP router-re mennek a csomagok. Ez a router a csomagokat még három router-nek továbbítja, mielőtt a csomag elérkezik a munkahelyi gépre. Tehát rengeteg gépen ment keresztül. A cracker-ek be tudnak törni a gépünkbe, a kapcsolódás másik végére (ami a telnet szervere), vagy a sok host valamelyikébe, ami közbe van iktatva.

Ha egyszer egy cracker átveszi az irányítást egy gép felett, képes lehallgatni minden olyan forgalmat a hálózaton, amiről a gépnek tudomása van. Ez felölel minden olyan csomagot, ami beérkezik, vagy elhagyja ezt a gépet. Ha a cracker gyökeret ver egy host-on, ami egy hub-on osztozik a célszerverünkkel, le tudja hallgatni a teljes session-t. Látni fogja, ahogy begépelem a nevem és a jelszavamat, és látni fogja a teljes rendszerben lévő neveket és jelszavakat, ahova onnan a telnet-en eljutok. Valójában azok a "lehallgató" eszközök, amiket erre használnak, rendszerint arra vannak konfigurálva, hogy egyszerűen fogjanak el annyi név/jelszó párost, amennyit csak lehet. De nem csak ez a probléma a telnet-el.

A további probléma a telnet-el a következő: mivel nincs rejtjelezés a két kapcsolódó gép között, egy cracker át tudja venni az irányítást a kapcsolat felett. Létezik egy jól ismert, könnyen kezelhető hunt-nak nevezett eszköz, ami ezt egyszerűvé teszi. A cracker az egyik csomagokat küldő gépen dolgozva lekapcsolhat minket és átveheti a helyünket a kapcsolatban, vagy leválaszthatja a szervert és átveheti annak a szerepét, vagy akár mindkettőt is megteheti egyszerre. Ez az eszköz sok kárt tehet.

Lássuk az ftp-t, a cracker-ek kedvenc célpontját!

Az ftp-nek nagyon sok hibája van. Rengeteg visszaélést lehet az ftp-n keresztül elkövetni. A legfontosabb először is az ftp tisztán szöveges, akárcsak a telnet. A cracker-ek kihasználva ezt, neveket és jelszavakat lopnak addig, míg mi végül meg nem szüntetjük a nem anonim ftp-t. Másodszor mivel az ftp szerverek rendszerint root-ként futnak, ez néhány nagyon is kézenfekvő távoli root-veszélyeztetést eredményezett. Ha ez a két ok nem elég, tessék egy újabb: az ftp a



legtöbb tűzfallal összegabalyodik!

Majdnem mindenhol telepítenek egyszerű csomag szűrő tűzfalakat, ami a legtöbb router-be be van építve vagy telepíthető a Linux-al. Ezért ha az ftp kliens (hagyományos) "aktív" módban fut, igen nagy port területet kell megnyitni a kliens host-okon. Ezt csökkenthetjük ha az ftp passzív, a passzív mód nem praktikus egy vállalkozás-szintű szervezetnél mert sokba kerül a felhasználók betanítása. Még ha mi passzív módba is kapcsoljuk a klienseket, ez azt kívánja meg a szerver tűzfaltól, hogy jókora port területet nyisson meg. Továbbá egy cracker ráveheti az ftp szervert arra, hogy inkább egy harmadik félnek adjon át adatokat, ahelyett, hogy visszaküldje azokat a klienseknek.

Rsh, Rcp és Rhosts-alapú Rlogin

Ezek szintén tisztán szöveges protokollok, de nem ez a fő oka annak, hogy nem biztonságosak. Ahogy a telnet-el és az ftp-vel használjuk őket, még rosszabb lesz a helyzet! Mindegyik az "rhosts" hitelesítést használja. Ez a hitelesítési séma nem biztonságos host-host trust -ot használ. A felhasználó egy "rhosts" állományt használva mondja meg azt, hogy a "root@sysadmin_box" root-ként beléphet a szerverre. A Host "szerver" ellenőrzi a csomagok forrás címét a hitelesítéshez. A probléma az, hogy ez az eljárás hibásan azt feltételezi, hogy egy támadó nem képes megmásítani a csomagokat sem a forrás gépen, sem a szállítás alatt. Sajnos képes, egyszerű automatikus eszközöket használva. Ezek az eszközök "átverik" a forrás IP-t, ezzel törvénytelen belépést engednek meg neki a cél host-ra. Nyilvánvaló, hogy az rhosts hitelesítésse nem megbízható, mert bármely olyan program át tudja verni, amely képes írni a csomagok fejlécét, és még azért is, mert bizalmon alapuló web-et hoz létre. Ha be tudom csapni "A" gépet úgy, hogy azt feltételezze, hogy egy olyan host gépen vagyok, amelyben megbízik, akkor az "A" gépre root-ként be tudok jelentkezni. Szóval ha "B" gép ugyanígy bízik "A" gépben, egyszerűen belépheték "B"-re. A cracker-ek gyakran visszaélnek a kiterjedt bizalmon alapuló web-el.

Egy cracker az egyik csomagokat küldő gépen dolgozva lekapcsolhat minket és átveheti a helyünket a kapcsolatban, vagy leválaszthatja a szervert és átveheti annak a szerepét, vagy akár mindkettőt is megteheti egyszerre.

A megoldás

A biztonsági héjjal (ssh) nincsenek ilyen problémák, helyettesítheti ezen protokollok mindegyikét.

Mivel a telnet tisztán szövegen és jelszón alapuló protokoll, amely a TCP-n működik, a támadó átkutathatja a session-t a jelszavakért, vagy átveheti az irányítást a session fölött. Az ssh nem támadható meg egyikkel sem, mivel rejtjelezőt alkalmaz, hogy elrejtse az egész session-t, a jelszavakat meg minden egyebet. Ha egy támadó figyeli a session-t, nem láthatja a jelszavakat vagy az adatokat - továbbá nem lehet részese a session-nek, mert nem rendelkezik a titkosító kulccsal. Nagyon leegyszerűsítve, ezt a kulcsot egy megegyezés szerinti public/private kulcs alapján lecserélik: a kliens generál egy session titkosító kulcsot, rejtjelezi a szerver nyilvános kulcsával és átküldi. A szerver visszafordítja ezt a kulcsot, mivel rendelkezik a privát kulccsal. Ettől kezdve mindkét gép ezzel a titkos kulccsal rejtjelezi a forgalmát. Egy közbenső személy nem tudja használni ezt az információt arra, hogy megszerezze a titkos kulcsot, mert nincs birtokában a szerver privát kulcsa.

Kulcspárok generálását az ssh-keygen parancsal kezdeményezhetjük. A kulcspár előállítása után egy jelszót kell beírnunk a privát kulcs védelme érdekében. Ezután a nyilvános kulcs pedig kiírásra kerül a képernyőre. A jövőben, amikor az ssh-nak szüksége lesz a privát kulcsra kérni fogja a jelszavunkat.

Állományok másolására az rcp helyett az scp parancsot használhatjuk. Például:

```
scp helyi_file
cél_host:elérési_útvonal
```

Ekkor az ssh kérni fogja a jelszavunkat:

```
cél_host's password:
```

A jelszó megadása után a "helyi file" átmásolódik a távoli gépre, de ebben az esetben a két

gép közötti kapcsolat már kódolt.

Az r-protokollok a csomagban lévő IP címet használják a hitelesítéshez, így "A"-host megbízhat "B"-hostban, úgy hogy a "B" hoston a sysadmin (root) automatikusan root lesz az "A" host-on. Ez könnyen kezelhető, de ezzel könnyen vissza lehet élni. Az ssh ezt helyettesíteni a következő képpen.

Az "A" host bízni akar "B" hostban. Azt tudnia kell, hogy a bejövő kapcsolat valóban a "B" host-ból van, nem hamis. Nos ha rendelkezik "B" public kulcsával, a következőt teszi. Először "A" létrehoz egy véletlen számot, titkosítja "B" public kulcsával, és elküldi "B"-nek. Csakis "B" lesz képes visszaalakítani ezt a számot, mivel csak "B" van birtokában a megfelelő privát kulcsnak. Aztán "B" egy előre meghatározott módon megváltoztatja ezt a számot, és visszaküldi "A"-nak. Ha "A" a helyes számot kapja vissza, akkor tudja, hogy "B"-vel kommunikál. Ezt nevezik RSA hitelesítésnek.

Nyilvánvaló, hogy az rhosts hitelesítésse nem megbízható, mert bármely olyan program át tudja verni, amely képes átírni a csomagok fejlécét

Míg a csomagok szempontjából ez biztonságos, valójában a host-ok közötti (host-to-host) bizalom tulajdonképpen mégsem a legjobb modell a mai számítástechnika számára. Létezik a nem jelszavas hitelesítésnek egy másik változata is, ami az account host bizalmon alapul. Míg felhasználja az RSA-alapú host ellenőrzést, le is ellenőrzi az aktuális felhasználó azonosságát. Ez megakadályozza, hogy "B"-hoston a root bármely felhasználót könnyen megszemélyesítse n. Vegyünk egy shost (biztonságos rhost) hitelesítést. Úgy állítjuk be a dolgokat, hogy "bob@host_A" be tud jelentkezni "bob@host_B"-be jelszó nélkül. De mi van ha valaki betör az "A" host root-jába? Vissza tud élni a host-ok közötti bizalommal úgy, hogy bob-ként jelentkeznek be a "B" host-on. Az RSA hitelesítés helyes használata esetén képes megállítani az ilyesfajta támadást. Vizsgáljuk meg a felhasználó-alapú RSA hitelesítést. Ez is az RSA nyílvános/privát kulcsú aszimetrikus titkosításon alapul, de ez inkább egy személyes kulcspártól függ, mint a host kulcspárjától. Lényegében a következőképpen működik:

Az account-unkról a forrásgépen hozzunk létre egy nyílvános és egy privát kulcs párt az "ssh-keygen" használatával. Ez létrehozza a privát kulcsot, ~/.ssh/identity, és a hozzátartozó nyílvános kulcsot, ~/.ssh/identity.pub.

Levédjük a privát kulcsot, és titokban tartjuk. Soha ne helyezzük át a fájlt a forrás-gépen lévő account-ról, különösen a hálón keresztül nem.

Adjuk hozzá a public kulcsot egy speciális fájlhoz, ~/.ssh/autherized_keys, a cél host-on lévő account-ban.

Amikor a forrás-gépen lévő account-ból beírjuk az "ssh user@target" parancsot, a cél host-on engedélyezik a belépést a user account-ba. A session-t részletesebben is megtekinthetjük a következő beírásával: "ssh -v user@target". A működési alapelv ugyanaz, mint az előbb. Az authorized_keys fájl minden egyes olyan account számára egy egyszerű (hosszú) sorral rendelkezik, ahonnan én kapcsolódhatok. A sor vége, megjegyzésként, még az account nevet is kiírja. Amikor futtatjuk az ssh-t, ez megkérdezi az sshd szervert, hogy elfogadja-e a ~/.ssh/identity.pub kulcsot a hitelesítéshez. Ha igen, a szerver egy véletlen számot generál, titkosítja a public kulccsal és elküldi a kliensünknek. Ha a kliens rendelkezik a megfelelő privát kulccsal, ~/.ssh/identity, akkor vissza tudja fejteni a véletlen számot, alkalmazza az előre meghatározott átalakítást és visszaküldi azt a szervernek. Amikor a szerver fogadja ezt a (helyes) számot, tudja, hogy rendelkezünk a titkos kulccsal és így engedélyezi a belépést. A jelszavak egyáltalán soha nincsenek átküldve a hálózaton, így nincs lehetőség a titkosítás-alapú (crypto-based) vagy a visszajátszó (replay) támadásokra. Végül ha megfeleően gondoskodunk a titkos kulcsról, az meglehetősen biztonságban van.

Ingyen hírdethet!

Díjmentesen megjelentetjük hírdetését ha

- munkát kínál
- keres vagy elad használt PalmTop-ot

Apróhirdetés jelleggel egyéb hirdetéseket is kedvezményesen jelentetünk meg (pl. honlapkészítés, szoftver készítés, stb...)

Továbbá kérheti hírdetését borítóra vagy belső oldalra színesben is.

Részletekért keresse meg kiadónkat,
vagy hívja munkatársunkat: 30/9386-875.

Legyen az Öné!

részletek a 18. oldalon